REMARKS

Claims 1-8 are pending in the application. No new matter has been added. At the request of the Office Action, a clean version of the claims is attached hereto.

Amendments to the claims clarifying that the computing section creates CAD part graphic data based on the basic data are supported in various places throughout the specification, such as in paragraphs 66 and 67 of the specification:

> The foregoing processing operations shown in the flowcharts of FIG. 2 to FIG. 4 are summarized as follows. In (steps S1 to S8), when the user selects parts, the server computer 1 transmits to the client computer 2 variable program and real data (parts data list) which is base data of the graphic data corresponding to the parts.
> Next, in steps S9 to S17, it substitutes real-number data that corresponds to the code number selected by the user into the variables of the variable program file that corresponds to the part selected by the user, and while interpreting the variable program by the interpreter-type programming language of the computing section 26, performs computation and creates graphic data based on the computation results, after which it creates display data to be displayed on the client computer 2 and displays that data.

Amendments to the claims clarifying that the variable program is created using a non-compiler interpreter-type programming language are supported in various places throughout the specification, such as in paragraph 261 of the specification:

> Also, the variable program of this invention is created in an interpreter-type programming language and does not rely on compiling techniques or other development languages, so it is possible to handle the variable program like data. The variable program exists in the overall CAD system like data, so it is possible to easily put it into a database. Moreover, when creating the variable program, by simply having knowledge of geometry and machine or architectural drawings, it is possible for anyone to easily create the program without having a special understanding of special software development such as compilation technology or debugging methods. It is also possible to concentrate on the contents to create. By doing so, it is possible to reduce costs when creating the variable program.

Double Patenting

Claims 1-8 have been rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over claims 1-5 of U.S. Patent No. 7,218,979.

A terminal disclaimer is submitted herewith to obviate the double-patenting rejection. Applicants, therefore, request that the double patenting rejection be withdrawn.

Claim Rejections - 35 U.S.C. § 102

The Office Action at page 7 rejects claims 1-8 under 35 USC 102(b) as being anticipated by Saha et al. ("Web-Based Distributed VLSI Design," hereafter "Saha"). The Office Action at page 7 rejects claims 1-8 under 35 USC 102(b) as being anticipated by Geppert et al. ("IC Design on the World Wide Web," hereafter "Geppert"). For the reasons described below, neither Saha, nor Geppert, describe each and every element of independent claims 1 or 2.

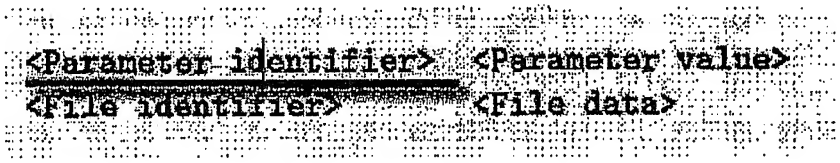
Claims 1 and 2 include the features of a computing section that creates CAD part graphic data based on basic data. Additionally, claims 1 and 2 include a CAD part graphic data producing section that creates display data for the graphic display unit in the client computer based on the CAD part graphic data created by the computing section. In contrast, Saha concerns a Web-based distributed VLSI (Very Large Scale Integration) design, which is the process of creating integrated circuits by combining thousands of transistor-based circuits into a simple chip. VLSI systems require distributed design and verification methodology due to the diverse expertise required at various areas. The system requires tools and information which is accessible through the Internet. Also in contrast, Geppert describes an IC Design method in which the parts are searched using many methods such as search engines. The parts found are for ordering. In addition it has the options to allow checking for availabilities. This clearly shows that the parts stored are found in order to place orders.

The Office Action asserts that Saha describes "current versus time" which is parametric graphic data. Claims 1 and 2 include CAD part graphic data being generated from the variable program which is unlike the Saha 'current versus time' system, because the claimed features generate CAD vector data for parts. For example, the real number data can be inputted by the

client beforehand and substituted in its corresponding variable program, to create the CAD part graphic data and the parts number.

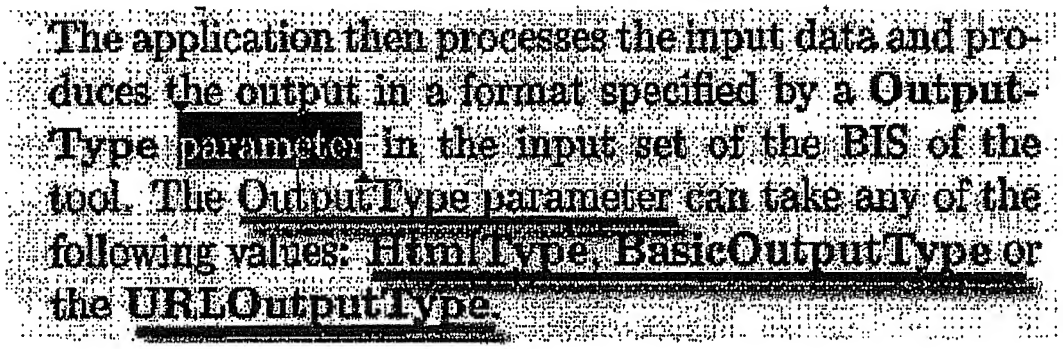
The Office Action asserts as follows:

iv) Applicants argue neither Saha nor Geppert disclose a parametric method, top of page 9 of Applicants remarks. The parametric method can be seen in Saha on page 451 bottom:



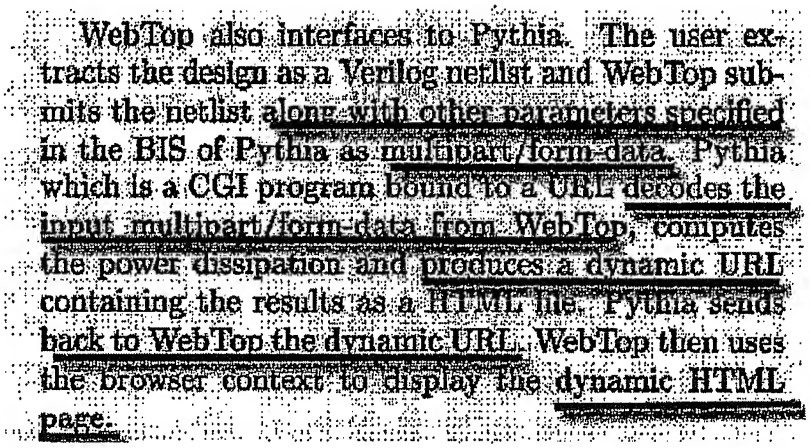
However, the method of Saha does not describe its application, where as claims 1 and 2 are used to build CAD data for parts. The Office Action also asserts that the parametric method is described as follows in Saha:

As well as page 452, left:



This is not the data produced or used to create CAD data for parts as in claims 1 and 2. The Office Action then asserts that the parametric method is described as follows in Saha:

As well as page 453, right:

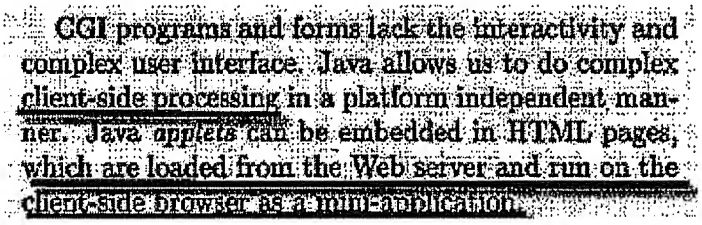


WebTop also interfaces to Pythia. The user extracts the design as a Verilog netlist and WebTop submits the netlist along with other parameters specified in the BIS of Pythia as multipart/form-data. Pythia, which is a CGI program bound to a URL decodes the input multipart/form-data from WebTop, computes the power dissipation and produces a dynamic URL containing the results as a HTML file. Pythia sends back to WebTop the dynamic URL. WebTop then uses the browser context to display the dynamic HTML page.

While this can display a dynamic HTML page, it does not generate a specific HTML according to its parameters, let alone generating CAD parts graphic data as in claims 1 and 2.

Additionally, the Office Action asserts as follows:

This is in response to Applicants arguments on the top of page 10. This can also be seen in Saha on page 450:

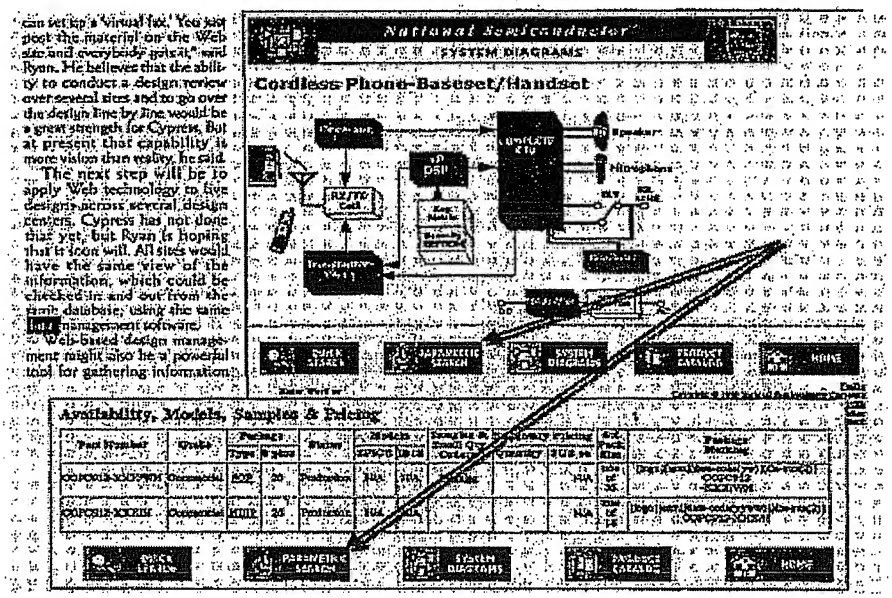


CGI programs and forms lack the interactivity and complex user interface. Java allows us to do complex client-side processing in a platform independent manner. Java *applets* can be embedded in HTML pages, which are loaded from the Web server and run on the client-side browser as a mini-application.

Saha does describe Java applets, however, it does not describe the use of a parametric method to create the CAD parts graphic data. Further, the JAVA applet is embedded in the HTML pages, therefore the system cannot send the required data from the database to a client computer using the web. This further indicates that the graphical data is not created in the client computer. In claims 1 and 2, a plurality of variable programs is stored in the variable program storage section, which is collected and sent to the client computer over the network.

With respect to Geppert, the Office Action notes as follows:

With respect to Geppert, see Figure 5 showing parametric data:



However, the below portion of Geppert makes clear that the parameters are computed to search for parts information. Geppert does not compute parameters to create CAD parts graphic data over a network, such as using the aid of the Internet.

Availability, Models, Samples & Pricing

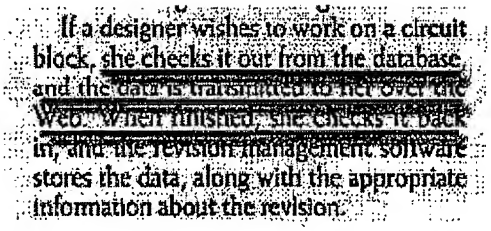
Part Number	Grade	Package Type	Package # pins	Status	Models SPICE	Models IBIS	Samples & Small Qty Orders	Budgetary Pricing Quantity	Budgetary Pricing $US ea	Std Pack Size	Package Marking
COPC912-XXXIWM	Commercial	SOP	20	Production	N/A	N/A	Samples		N/A	tube of 36	[logo][asm][date-code(yyww)][die-run(2)] COPC912 -XXXIWM
COPC912-XXXIN	Commercial	MDIP	20	Production	N/A	N/A	.		N/A	tube of 18	[logo][asm][date-code(yyww)][die-run(2)] COPC912-XXXIN

QUICK SEARCH PARAMETRIC SEARCH SYSTEM DIAGRAMS PRODUCT CATALOG HOME

[5] One of the search options on the Web site of Natibnal Semiconductor Corp., Santa Clara, Calif., is the system diagram [top]. Clicking on any of the components shown brings up a list of part numbers that could serve as that component. Selecting any part from the list brings up a table with more information about the part and its availability.

The Office Action further relies on the following from Geppert:

Further see Geppert page 49:



This shows the data is transmitted to the client, is worked on client side, and then sent back to the server.

However, Geppert utilizes the Web to retrieve the required data and transmit it to a data base where it is saved. Geppert does not use the Web to transmit information to create CAD parts graphic data for display purposes.

The Office Action also asserts:

> v) With respect to the interpreter type programming language the Examiner notes that interpreters and compilers are two well known methods for the implementation of a programming language. They are also not mutually exclusive. For example, with respect to Java, source code is compiled and then linked at runtime and executed by an interpreter such as a Java Virtual Machine, DynamicJava, and BeanShell. CORBA is also capable of being run through an interpreter such as CorbaScript and GSCRIPT. Both CORBA and JAVA are programming languages which can be both interpreters/compiled type based on their runtime environment and as such the rejection is maintained.

However, with respect to the interpreter-type programming language of claim 2, it can create the variable program. Further, the variable program can be sent to the client computer and used to create the CAD parts graphic data.

Claims 3-8 depend from claims 1 and 2, respectively, and thus are also patentable over the cited art.

The present invention as claimed is, therefore, believed to be patentable over the art and the Examiner is requested to withdraw the rejections under 35 USC 102.

Favorable consideration and early issuance of the Notice of Allowance are respectfully requested. Should further issues remain prior to allowance, the Examiner is respectfully

requested to contact the undersigned at the indicated telephone number.

Respectfully submitted,

/Andrew Gust/

Date: September 22, 2008

Andrew Gust (Registration No. 47,620)
Yonghong Chen (Registration No. 56,150)
Akerman Senterfitt
Customer No. 30448
222 Lakeview Avenue, Suite 400
West Palm Beach, FL 33401
Phone: 561-653-5000

CLEAN VERSION OF CLAIMS

1. (currently amended) A CAD part library generator system utilizing a network and comprises:

a server computer that is connected to a network; and

at least one client computer that performs data transmission with said server computer via said network;

said server computer sends basic data, which are combinations of plurality of variable programs for drawing different part graphics and numerical data that are substituted into the variables of said variable programs, for CAD part graphic data from said server computer to said client computer according to a request from said client computer;

wherein said server computer comprises:

a storage means that stores basic data for said CAD part graphic data; and

a program data transmitting section that reads said basic data for CAD part graphic data from said storage means according to a request from said client computer, and sends that data to said client computer;

said client computer comprises:

a program data receiving section that receives said basic data for CAD part graphic data;

a computing section that creates CAD part graphic data based on said basic data; and

a CAD part graphic data producing section that creates display data for the graphic display unit in said client computer based on the CAD part graphic data created by said computing section;

said storage means of said server computer comprises a variable program storage section that stores said plurality of variable programs, and a numerical data storage section that stores a plurality of kinds of said numerical data according to a request from said client computer, then sends the specified variable and numerical data to said client computer; and

said computing section of said client computer substitutes said numerical values of specified numerical data into the variables of said specified variable program, then executes that program and creates CAD part graphic data.

2. (currently amended) A CAD part library generator system utilizing a network and comprises:

a server computer that is connected to a network; and

at least one client computer that performs data transmission with said server computer via said network;

said server computer sends basic data for CAD part graphic data from said server computer to said client computer according to a request from said client computer;

wherein said server computer comprises:

a storage means that stores basic data for CAD part graphic data; and

a program data transmitting section that reads said basic data for CAD part graphic data from said storage means according to a request from said client computer, and sends that data to said client computer;

said client computer comprises:

a program data receiving section that receives said basic data for CAD part graphic data;

a computing section that creates CAD part graphic data based on said basic data for CAD part graphic data; and

a CAD part graphic data producing section that creates display data for the graphic display unit in said client computer based on the CAD part graphic data created by said computing section;

said basic data for CAD part graphic data comprises a plurality of variable programs for drawing different part graphics and numerical data that is substituted into the variables of said variable programs;

said storage means of said server computer comprises a variable program storage section that stores said plurality of variable programs, and a numerical data storage section that stores a plurality of kinds of said numerical data;

said program data transmitting section reads a specified variable program from said variable program storage section, and reads specified numerical data from said numerical data storage section according to a request from said client computer, then sends that data to said client computer;

said variable program is created using a non-compiler interpreter-type programming language; and

said computing section of said client computer comprises an interpreting function of said non-compiler interpreter-type programming language, and substitutes said specified numerical data into the variables of said specified variable program, then executes that variable program while interpreting it by the interpreting function of said computing section, and creates CAD part graphic data.

3. (currently amended) The CAD part library generator system utilizing a network according to claims 1 or 2 wherein said client computer further comprises:

a graphic name list display control section for displaying a list of received graphic names of the basic data for CAD part graphic data provided from said server computer on the display unit; and

a selected graphic name transmitting section that sends the names of graphics selected from said list of graphic names to said server computer;

said program data transmitting section in said server computer reads said specified variable program and specified numerical data based on the graphic names that were sent from said selected graphic name transmitting section.

4. (currently amended) The CAD part library generator system utilizing a network according to claims 1 or 2 wherein said server computer further comprises:

a parts data list storage section that groups and stores part code numbers for each part and said numerical data corresponding to the code numbers;

said program data transmitting section transmits the part data list containing the code numbers and the numerical data to said client computer according to a request of said client computer;

said client computer further comprises:

a code number list display control section that creates a parts code number list from said sent parts data list transmitted, and displays the list on said graphics display unit; and

said computing section substitutes numerical data for the parts that correspond to the names of the part code numbers selected from said displayed parts code number list into the variables of the variable program that corresponds to the names of said graphics and creates CAD part graphic data.

5. (currently amended) The CAD part library generator system utilizing according to claim 4 wherein when part or all of the numerical data selected by the user in said client computer corresponds to the part code numbers selected from said part code number list in said client computer, said computing section of said client computer substitutes said numerical data that was read from said parts data list storage section and said input data into the variables in said corresponding variable program and creates CAD part graphic data.

6. (currently amended) The CAD part library generator system utilizing a network according to claims 1 or 2 wherein said client computer further comprises:

a data format name selection function that is capable of selecting a data format name for the CAD software; and

said CAD part graphic data producing section converts the format of the CAD part graphic data created by said computing section, creates CAD part graphic data that suits the selected data format, assigns a file name and stores the data in the memory unit.

7. (currently amended) The CAD part library generator system utilizing a network according to claims 1 or 2 wherein said client computer further comprises:

an interface name selection function that is capable of selecting a name for the data-exchange interface provided by the CAD software; and

said CAD part graphic data producing section converts the format of the CAD part graphic data created by said computing section to create CAD part graphic data, and registers said CAD part graphic data directly in said CAD software by way of said data-exchange interface.

8. (currently amended) The CAD part library generator system utilizing network according

to claims 1 or 2, further comprising a parts database management program for managing parts data in said program data transmitting section of said server computer.